

---

This is the **published version** of the bachelor thesis:

Roldán Vilardell, Laura; Espinosa Morales, Antonio, dir. Reconeixement facial amb models Deep Learning a la plataforma Amazon DeepLens i AWS. 2021. (958 Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/248461>

under the terms of the  license

# Reconeixement facial amb models Deep Learning a la plataforma Amazon DeepLens i AWS

Laura Roldán Vilardell

29 de juny del 2021

**Resum**– L'autenticació de l'usuari és un procés que fa anys que es posa en pràctica. Ofereix seguretat a un sistema d'accés i dona privilegis a certs usuaris. Tanmateix, a causa de la COVID-19, molts d'aquests sistemes han hagut de renovar-se i oferir una seguretat sanitària pels empleats. Aquest projecte utilitza les noves tecnologies del Cloud Computing i Deep Learning per realitzar un sistema d'accés a un edifici restringit utilitzant reconeixement facial.

Utilitzant la càmera d'Amazon DeepLens es procedeix a fer una detecció de la cara, i amb els serveis de visió per computador, es duen a terme les funcionalitats de detecció de mascaretes i reconeixement de l'individu. A més a més, l'usuari rep un *feedback* per veu durant tot el procés d'autenticació al sistema.

**Paraules clau**– Amazon Web Services, Cloud, Deep Learning, DeepLens, reconeixement facial, visió per computador

**Abstract**– User authentication is a process that has been in practice for years. It provides security to an access system and gives privileges to certain users. However, due to COVID-19, many of these systems have had to be updated and provide health security for employees. This project uses new Cloud Computing technologies and Deep Learning to implement a restricted building access system using facial recognition.

Using the Amazon DeepLens camera, a face detection model is executed, and with the computer vision services, the mask detection and facial recognition functionalities are performed. In addition, the user receives audio feedback throughout the process of authentication to the system.

**Keywords**– Amazon Web Services, Cloud, computer vision, Deep Learning, DeepLens, facial recognition

## 1 INTRODUCCIÓ - CONTEXT DEL TREBALL

A causa de la situació actual d'emergència sanitària, ens trobem obligats a portar mascareta, evitar el contacte el màxim possible i netejar-nos les mans cada cop que toquem un objecte d'ús públic. Per la qual cosa, molts sistemes d'ús comú i quotidians s'han hagut de reinventar per ser acceptats per les noves restriccions.

En aquest projecte es proposa una solució per millorar el

sistema d'accés a un edifici restringit, concretament el del Centre de Visió per Computador. Per aconseguir aquest objectiu, s'utilitzen eines d'intel·ligència artificial i visió per computador que aporta el Cloud d'Amazon (Amazon Web Services).

Principalment es vol dissenyar un sistema d'accés a un edifici mitjançant reconeixement facial, superant el repte de les normatives sanitàries imposades per la COVID-19.

## 2 OBJECTIUS

Després d'analitzar la problemàtica, es defineixen una sèrie d'objectius a complir, per concloure el projecte amb èxit:

- Millorar la seguretat per la salut

---

• E-mail de contacte: laura.roldanvi@e-campus.uab.cat  
• Menció realitzada: Computació  
• Treball tutoritzat per: Antonio Espinosa Morales (Departament d'Arquitectura de Computadors i Sistemes Operatius)  
• Curs 2020/2021

- Minimitzar el temps. Simplificar tot el procés d'accés
- Digitalitzar el sistema i fer-lo escalable
- Millorar la seguretat d'accés. Evitar una possible suplantació d'usuari.

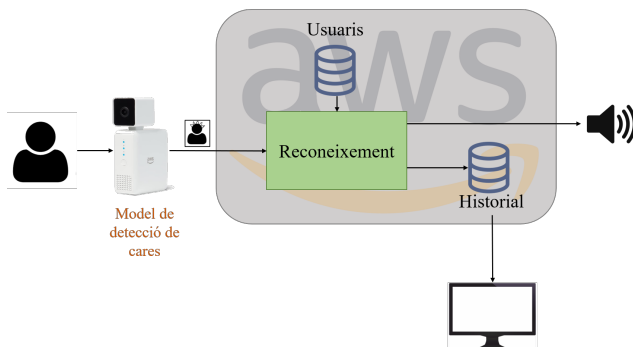


Fig. 1: Proposta de la solució

Per assolir-los es proposa dissenyar un sistema com es mostra en la figura 1, seguint l'exemple del projecte referenciat a [1]. L'element principal del disseny és l'**AWS DeepLens** [2]. Aquest es tracta d'una càmera de vídeo amb capacitat d'executar models d'aprenentatge profund en local. Està integrada amb diversos serveis d'aprenentatge automàtic del núvol d'Amazon i permet executar models creats dins d'AWS. Amb aquesta càmera es capten les cares dels usuaris. A més, es fa la crida a un model de detecció de mascaretes per comprovar que l'individu a autenticar la porti posada, en cas contrari, se li denega l'entrada fins que se la posi.

Les especificacions tècniques més importants són les següents [3]:

- La CPU és un processador Intel Atom
- La memòria RAM és de 8 GB
- L'emmagatzemament (disc) és de 16GB, però permet expansió
- El sistema operatiu és Ubuntu OS-16.04 LTS
- La càmera és de 4 megapíxels amb MJPEG
- Connexió a Internet via Wi-Fi (2.4 GHz i 5GHz) o Ethernet
- Té sortida d'àudio i micròfon
- Dispositiu IoT de baix consum (20 Watts) de potència

A continuació, es mostra una caixa negra, que representa el núvol. Allà es procedeix a comparar la imatge captada amb el registre d'usuaris de la base de dades, per així poder identificar l'individu. A més, es guarda el registre d'entrades.

Finalment, hi ha l'*output*. En el cas que el reconeixement hagi estat satisfactori, a través d'àudio s'escolta el *feedback* del procés i el nom de la persona. En cas contrari, denega la possibilitat d'accedir a l'edifici i emet un missatge d'assistència tècnica.

El sistema incorpora un element de monitoratge, que des d'una aplicació web, permet observar quina és l'última persona que s'ha autenticat.

### 3 ESTAT DE L'ART

En l'actualitat, diverses empreses estan usant mètodes biomètrics com a autenticació dels seus empleats. L'objectiu és crear una experiència més dinàmica per l'usuari, ja que no requereix cap element físic (com un PIN o una targeta), millorar la seguretat, i evitar que el sistema pugui ser enganyat. Tot i això, aquests mètodes s'estan aplicant a diferents sectors, des de supermercats, fins a escoles.

El mètode d'autenticació per reconeixement facial es porta estudiant des de fa anys, i ha donat com a resultat la creació de softwares com FacePRO, FaceFirst, entre d'altres [4].

A la investigació i millora d'aquesta tecnologia, empreses de màxim exponent tecnològic (GAFAM - Google, Amazon, Facebook, Apple, Microsoft) continuen desenvolupant aquest mètode per aconseguir un sistema més ràpid i precís [5].

Al mateix temps, amb la nova aparició dels serveis Cloud Computing permet que aquesta tecnologia estigui a l'abast de tothom gràcies a la facilitat de llogar serveis en comptes de posseir un gran servidor local. Els principals proveïdors Cloud són Amazon (Amazon Web Services), Microsoft (Microsoft Azure), Google (Google Cloud Platform) i IBM (IBM Watson). A la figura 2 es mostra una comparativa entre les capacitats dels seus serveis de visió per computador i aprenentatge computacional.

IMAGE ANALYSIS APIs COMPARISON				
	Amazon	Microsoft	Google	IBM
Object Detection	✓	✓	✓	✓
Scene Detection	✓	✓	✓	✗
Face Detection	✓	✓	✓	✓
Face Recognition (person face identification)	✓	✓	✗	✗
Facial Analysis	✓	✓	✓	✓
Inappropriate Content Detection	✓	✓	✓	✓
Celebrity Recognition	✓	✓	✓	✗
Text Recognition	✓	✓	✓	✓
Written Text Recognition	✓	✓	✓	✗
Search for Similar Images on Web	✗	✗	✓	✗
Logo Detection	✗	✗	✓	✗
Landmark Detection	✗	✓	✓	✗
Food Recognition	✓	✓	✗	✓
Dominant Colors Detection	✗	✓	✓	✗

Fig. 2: Comparació de l'anàlisi d'imatge dels quatre Clouds [6]

## 4 METODOLOGIA I PLANIFICACIÓ

La metodologia a utilitzar per a dur a terme aquest projecte, s'ha d'adaptar, per una banda, a l'escassa experiència en l'ús d'aquestes tecnologies i per una altra, a l'elevat nombre de persones involucrades. És per això que es determina la implementació de la solució a través d'aproximacions iteratives, molt alineat a la metodologia Scrum.

Aquesta és una metodologia àgil basada en sprints. Aquests han tingut una durada entre una setmana o dues, depenent de les dificultats de la tasca. A més, les reunions han estat setmanals. L'estructura és mostrada a la taula 1.

TAULA 1: PLANIFICACIÓ DEL PROJECTE EN SPRINTS

Sprint 1	Investigació i recopilació d'informació
Sprint 2	Definició del projecte
Sprint 3	Anàlisi i realització de proves amb la DeepLens
Sprint 4	Anàlisi i realització de proves amb les funcions Lambda
Sprint 5	Solució de configuracions de la DeepLens
Sprint 6	Realització de proves unitàries de les APIs a utilitzar
Sprint 7	Realització del reconeixement de cares
Sprint 8	Realització de la funcionalitat de registre d'usuaris. Creació d'una base de dades d'historial
Sprint 9	Migració del projecte d'un compte Educate a un normal (usant AWS Startup)
Sprint 10	Realització de la integració amb veu
Sprint 11	Creació d'un fitxer log amb accés des d'Internet
Sprint 12	Realització del model de detecció de mascaretes
Sprint 13	Realització del model de detecció de cares
Sprint 14	Desenvolupament de l'informe final
Sprint 15	Desenvolupament del dossier i presentació finals

Per a la planificació, seguiment i control del projecte s'han utilitzat bàsicament dues eines: Microsoft Teams i Gantt Project.

S'utilitza l'entorn de col·laboració de Microsoft Teams com a eina de comunicació. En aquesta aplicació s'ha creat un equip amb tots els membres involucrats. A més a més, aquesta eina també aporta un taulell (tipus Kanban) per fer el seguiment de les tasques. Per complementar, té unes carpetes per anar afegint arxius de progrés de les feines realitzades.

S'ha dissenyat la planificació a partir d'una llista en *top-down* de fases, activitats i tasques. Per poder controlar el temps d'execució de cada tasca, s'utilitza l'eina del diagrama de Gantt, concretament el Gantt Project.

Per la part d'execució, el projecte està desenvolupat dins de l'entorn del Cloud d'Amazon: Amazon Web Services. Com que aquest servei és costós, s'ha sol·licitat crèdits, a través d'AWS Startup, per cobrir les despeses del desenvolupament del projecte i els serveis AWS associats.

## 4.1 Minimum Viable Product

Seguint la metodologia Scrum, el desenvolupament del projecte s'ha dut a terme de manera incremental. És a dir, l'objectiu és arribar a un producte mínim viable i anar afegint-hi noves funcionalitats al producte existent.

Seguint aquesta idea es van proposar tres mínims productes viables, que seran considerades com uns *milestones* del projecte i asseguraran les següents funcionalitats:

- Prototip reconeixement de la cara amb la DeepLens i l'API Rekognition
- Prototip amb integració de veu
- Integració prototip en entorn d'usuari
- Prototip amb detecció de mascareta

El primer MVP, consisteix en la captació de la cara a través de la càmera i aquesta és comparada en una base de dades per identificar la persona. La identificació es duu a terme mitjançant la invocació de l'API Rekognition, que permet fer anàlisis d'imatge i vídeo amb aprenentatge profund [7].

El següent MVP incorpora *feedback* per veu a la persona que vol entrar per la porta.

A continuació, s'integra l'entorn d'usuari incorporant un fitxer *log* que pot ser llegit des de qualsevol dispositiu amb Internet i un navegador, a través d'un enllaç.

Finalment, en el darrer MVP s'afegeix la funcionalitat de detectar i assegurar que la persona porti la mascareta posada.

En les següents seccions es mostra en més detall els passos que s'han seguit per aconseguir aquestes fites.

## 5 ARQUITECTURA

Per la solució del problema plantejat, es proposa l'arquitectura de la figura 3. En aquesta imatge es mostren cinc mòduls definits.

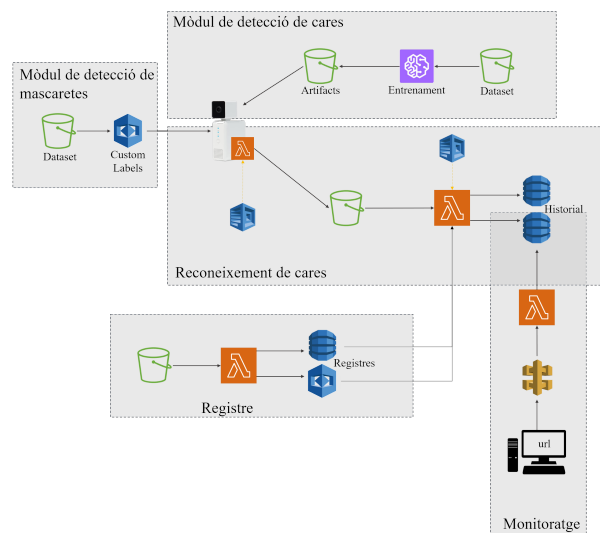


Fig. 3: Arquitectura de la solució

El mòdul de “reconeixement de cares” és el responsable de, com bé el nom indica, reconèixer l'individu un cop es posa davant la càmera. Aquest serà el mòdul principal en el flux de dades durant el funcionament del sistema.

El mòdul de “registre” és l'encarregat de registrar un nou individu. S'afegeix una imatge de la cara amb el nom de la persona a una base de dades.

El mòdul “monitoratge” permet observar des d'un enllaç al *log*, l'última persona que s'ha autenticat.

Cal tenir en compte, que abans de procedir al reconeixement de l'individu, primer s'ha de detectar la cara. Per fer aquest procediment es crea un model de detecció de cares i un model de detecció de mascaretes. Es necessiten dos models, ja que un cop es detecti la cara, s'ha de procedir a analitzar si la persona porta mascareta o no.

El mòdul de “detecció de cares” és on es preparen les dades i s'entrena el model utilitzant l'API Sagemaker, que a continuació, s'optimitza dins la DeepLens. Aquest model funciona de manera local a la càmera.

El mòdul de “detecció de mascaretes” és semblant a l'anterior, però en aquest cas s'utilitza l'API Rekognition per l'entrenament del model.

## 6 CONFIGURACIÓ EXPERIMENTAL

Seguint els objectius marcats s'ha desenvolupat el projecte, desglossant-lo en quatre etapes principals. D'aquesta manera es pot atacar al problema de manera individual.

1. Model de detecció de cares
2. Model de detecció de mascaretes
3. Detecció de cares amb la DeepLens
4. Integració de serveis AWS Cloud en entorns IoT

### 6.1 Model de detecció de cares

En un primer moment s'ha utilitzat el model preentrenat [8] que ofereix l'API d'AWS DeepLens. Tanmateix, aquest model s'ha descartat, ja que, encara que no hi hagi informació del dataset que utilitza, fent proves, s'ha deduït que està entrenada amb cares que no porten mascareta. Aquest fet perjudica molt la detecció, en aquest projecte, on s'obliga a l'individu a portar mascareta. Per la qual cosa, es proposa dissenyar un model amb l'eina Sagemaker utilitzant un dataset de cares amb mascareta i sense.

El Sagemaker és la peça central dels serveis de *machine learning* ofert per AWS. Busca facilitar el procés de desenvolupament de models, ja que en un mateix entorn es poden usar totes les eines i components per dissenyar un model d'aprenentatge computacional. D'aquesta manera, cobreix totes les etapes d'aquest projecte: creació del *groundtruth*, *training*, *tuning* i *deployment*. A través de la interfície visual d'Amazon SageMaker Studio es poden dur a terme tots aquests passos.

Per fer l'entrenament s'utilitza l'algorisme *Object De-*

*tection*. Aquest s'utilitza per detectar i classificar objectes en imatges. La localització de l'objecte s'indica amb una *bounding box* rectangular. Utilitza el mètode *Single Shot multibox Detector* (SSD), el *framework* mxnet, i una xarxa base ResNet-50 [9].

El dataset consisteix en 3.813 imatges de persones portant mascareta o no. Mitjançant la consola de *Ground truth* es creen els *bounding boxes* i s'etiqueten per a cada imatge.

Un cop les dades estan preparades s'ha disposat a realitzar diversos entrenaments, per veure quin és el conjunt d'hiperparametres més adients per aconseguir un bon resultat. Primer de tot, s'ha dividit el dataset en *training* i *test*, 90% i 10% respectivament.

A continuació, utilitzant la funció d'*hyperparameters tuning job* s'ha permès crear un conjunt d'entrenaments variant alguns hiperparametres assenyalats. A la taula 2 es mostren les opcions d'entrenament.

Els hiperparametres que es varien durant el *tuning* són:

- *learning\_rate*
- *mini\_batch\_size*
- *momentum*
- *weight\_decay*

L'objectiu és maximitzar la mètrica “*validation:mAP*”.

Per avaluar models de detecció d'objectes s'ha utilitzat *mean Average Precision* (mAP). El mAP compara el *groundtruth bounding box* amb el *box* detectat i retorna una puntuació. Com més alta sigui la puntuació, més precís serà el model en les seves deteccions.

És una manera de resumir la corba precision-recall en un únic valor, representant la mitja de totes les precisions.

Un cop finalitzat l'entrenament, els fitxers de sortida s'anomenen *artifacts*. Aquests consten d'un .json i un .params, que defineixen l'arquitectura del model. A més a més es guarda un .json pels hiperparametres.

### 6.2 Model de detecció de mascaretes

Una de les funcionalitats a afegir és la detecció de la mascareta, és a dir, la càmera ha d'analitzar si la persona que es vol reconèixer, porta mascareta posada o no.

La idea inicial, era crear un model amb Sagemaker, semblant a l'anterior, i fer un segon *deployment* d'ell a la càmera. D'aquesta manera s'aconseguiria tenir els dos models corrents de forma local. Tanmateix, a causa de les restriccions de la DeepLens, que no permet tenir més d'un model integrat, es va buscar un pla alternatiu.

Amazon Rekognition és una API del Cloud que facilita l'anàlisi d'imatge i de vídeo. Dins de totes les opcions que presenta aquesta eina, per fer el model de detecció se centra a les funcions de les etiquetes, concretament a la personalització d'etiquetes.

D'entrada, Rekognition té una sèrie de paràmetres prede-

TAULA 2: HIPERPARAMETRES SELECCIONAT PER L'ENTRENAMENT

Nom hiperparametre	Definició	Tipus	Valor/Rang
base_network	L'arquitectura de la xarxa base	Static	resnet-50
epochs	Nombre d'epochs	Static	100
freeze_layer_pattern	L'expressió regular (regex) per congelar capes de la xarxa base	Static	false
image_shape	Mida de les imatges d'entrada	Static	300
label_width	L'amplada de l'etiqueta de farciment de força que s'utilitza per a la sincronització entre dades d'entrenament i validació	Static	150
learning_rate	Taxa d'aprenentatge inicial	Continuous: Logarithmic	0,0001 - 0,05
lr_scheduler_factor	Proporció per reduir la taxa d'aprenentatge	Static	0,2
lr_scheduler_step	Les epochs en les que s'ha de reduir la taxa d'aprenentatge	Static	10 20
mini_batch_size	Mida del lot per l'entrenament	Integer: Linear	30 - 50
momentum	El momentum per sgd	Continuous: ReverseLogarithmic	0,0 - 0,99
num_classes	Nombre de classes a detectar	Static	1
num_training_samples	Nombre d'imatges d'entrenament	Static	3.432
optimizer	Tipus d'optimitzador	Static	sgd
use_pretrained_model	Indica si s'utilitza un model pre-entrenat per l'entrenament	Static	1
weight_decay	El coeficient weight decay per sgd	Continuous: Logarithmic	0,00001 - 0,99

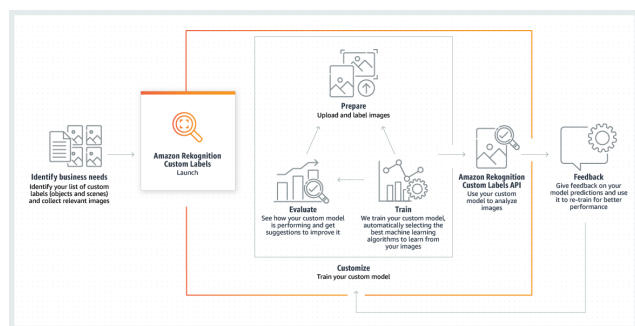


Fig. 4: Funcionament de la funció d'etiquetes personalitzables

terminats per fer l'anàlisi d'una cara. És capaç de detectar si es tracta d'una cara, si somriu o no, si té ulleres, etc. Però, a més a més, es poden crear aquests valors. Per la qual cosa, es crea una etiqueta de "mascareta": s'analitza la imatge per saber si una cara d'una persona porta mascareta o no.

Les etiquetes personalitzades d'Amazon Rekognition es basen en AutoML (Automated Machine Learning): Es redueix el temps necessari per crear models de machine learning.

Enlloc d'utilitzar milers d'imatges, simplement carregant un reduït conjunt d'entrenament a la consola de l'API, és suficient. En aquest cas el dataset està format per 1.376 imatges: 690 persones amb mascareta i 686 persones sense mascareta.

Etiquetar les imatges ha estat un procés automàtic: Cada conjunt de classe es troba en una carpeta diferent amb el nom de l'etiqueta que li pertocaria, i la mateixa eina fa aquest etiquetatge.

Una vegada que es comença l'entrenament, es pot produir un model d'anàlisi d'imatges personalitzat en poques hores. En segon pla, les etiquetes personalitzades de Rekognition

es carreguen i examinen automàticament les dades d'entrenament, seleccionen els algorismes adequats de machine learning, entrenen un model i proporcionen la mètrica de rendiment del model. A continuació, ja es pot utilitzar el model personalitzat i integrar-lo a les aplicacions.

### 6.3 Detecció de cares amb la DeepLens

Com s'ha comentat des de l'inici, l'element principal d'aquest projecte és la càmera DeepLens. La seva característica primordial és que es permet executar la inferència d'un model a la mateixa càmera i que té una integració molt senzilla amb el Cloud d'Amazon.

Destacar que la DeepLens és considerada un dispositiu de l'API IoT Greengrass, que permet l'enviament de missatges MQTT de la càmera al Cloud. MQTT és considerat el protocol més habitual en dispositius IoT, que permet l'interconnexió d'aquests des de qualsevol ubicació cap a un gestor centralitzat de gestió de missatges.

Finalment, comentar que aquesta càmera funciona de manera semblant a una de vigilància, pel fet que sempre estarà captant imatges.

Aquesta càmera serà l'element on es farà el *deployment* dels dos models comentats anteriorment. En el cas del model de detecció de cares se li afegeix una funció Lambda encarregada de cridar aquest model [11] en local. En l'altre cas, el model de detecció de mascaretes s'executa externament al Cloud.

Abans de poder fer correctament el desplegament del model, aquest s'ha d'optimitzar perquè la càmera el pugui executar. Per fer aquest pas, s'utilitza l'optimitzador de model Intel OpenVino. Aquest es tracta d'un conjunt d'eines per desenvolupar ràpidament aplicacions i solucions que resolen diverses tasques, com ara l'emulació de la visió humana, el reconeixement automàtic de la parla, el processament del llenguatge natural o els sistemes de recomanació [12].



Aquest paquet es troba preinstal·lat al sistema operatiu de la DeepLens, i és l'encarregat d'optimitzar i desplegar el model entrenat amb el *framework* mxnet [13]. Tanmateix, aquest programari és força restrictiu. Per això, algunes de les característiques del model amb les que s'han entrenat, han estat escollides perquè l'optimitzador funcionés. Aquest és el cas de la mida d'imatge d'entrada (només pot ser 300x300) i del nom de les capes de la xarxa (només són acceptades les proposades per l'optimitzador).

Un cop el model s'ha optimitzat correctament per les especificacions limitades, s'ha dut a terme el *deployment*.

## 6.4 Integració de serveis AWS Cloud en entorns IoT

El *Cloud Computing* [14] és una forma de computació que té els seus fonaments a Internet i que mitjançant la qual, els recursos compartits, el programari, i la informació, es proporcionen a la carta com a serveis. D'aquesta manera, un usuari lloga aquests serveis, i al estar treballant en un entorn remot, no té cap recurs en local que s'hagi de fer càrrec.

Aquest projecte és un exemple on es mostra el benefici que aporta l'ús del Cloud, ja que requereix tant un nombre elevat de recursos com de funcionalitats. Aquests reptes són possible assolir-los gràcies a les diferents APIs que Amazon Web Services proposa, per una banda, i, per una altra, a la possibilitat d'executar-les remotament, fent ús de recursos "quasi infinits".

El Cloud d'Amazon té un inventari molt estès de recursos que es poden sol·licitar, dels quals, per aquest projecte s'utilitzen la llista següent:

- Lambda: Permet crear i executar codi per fer les crides a les altres APIs
- Simple Storage Service (S3): Servei d'emmagatzemament d'objectes
- Rekognition: Facilita l'anàlisi d'imatge i vídeo
- DynamoDB: Servei de base de dades
- Polly: Servei que converteix text en parla
- API Gateway: Servei que permet crear, mantenir i protegir les APIs

La càmera DeepLens utilitza recursos externs només quan els necessita, no s'està executant una màquina virtual tota l'estona. D'aquesta manera, s'estalvia tenir una instància Cloud oberta en moments d'inactivitat, com per exemple de nit, període de temps on no hi haurà accessos a l'edifici.

Gràcies a les funcions Lambda permet crear un sistema asíncron, invocant als recursos quan arriba una nova persona que vol entrar a l'edifici.

La DeepLens treu el màxim partit d'aquestes característiques, ja que, d'una banda el procés del model de detecció de cares, s'executa localment, sense requerir recursos externs del Cloud, i per una altra banda, la funció Lambda

invoca als recursos Cloud únicament quan són requerits.

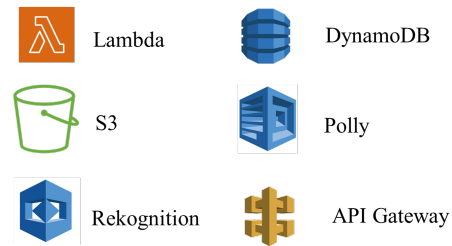


Fig. 5: Llegenda de les APIs

En la figura 5 es mostra la llegenda logo-API, dels serveis que es trobaran representats gràficament més endavant.

## 7 RESULTATS

A continuació s'exposen els diferents experiments realitzats, juntament amb els resultats obtinguts.

### 7.1 Resultats del model de detecció de cares

Principalment s'han executat 40 *training jobs* modificant els hiperparametres *learning\_rate*, *mini\_batch\_size*, *momentum* i *weight\_decay*. L'objectiu és maximitzar *mean Average Precision* (mAP). La definició general d'aquesta mètrica és trobar l'àrea de la corba precision-recall [15]. Els resultats es mostren en la figura 6, on la gràfica mostra els 40 entrenaments i quin és el mAP que han assolit cadascun d'ells.

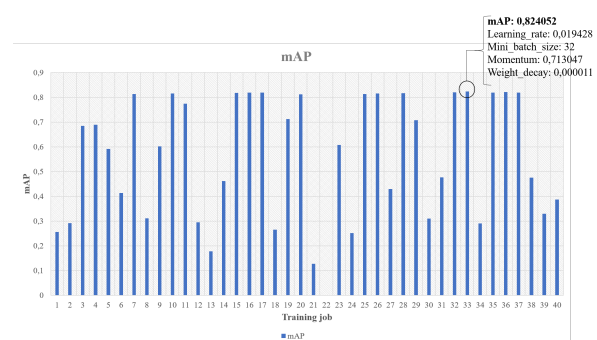


Fig. 6: Comparació de mAP entre tots els entrenaments

En la mateixa gràfica, es representa com l'entrenament número 32, és el que ha assolit un mAP més alt sobre la resta: 0,824052. L'entrenament ha durat 3.757 segons (1,0436h), utilitzant el tipus d'instància ml.p3.2xlarge, que consisteix en una màquina de 8 CPU virtuals i 61 GiB de memòria.

En la figura 7 es mostra el progrés de mAP a través del temps.

En la figura 8 es mostra les gràfiques on es representen les mètriques "train:cross\_entropy" i "train:smooth\_l1".

Aquestes dues mètriques representen dues funcions *loss*. Com valors més baixos tinguin, el model serà millor.

Els resultats obtinguts són prou favorables per la detecció d'una cara amb una persona portant mascareta o sense.

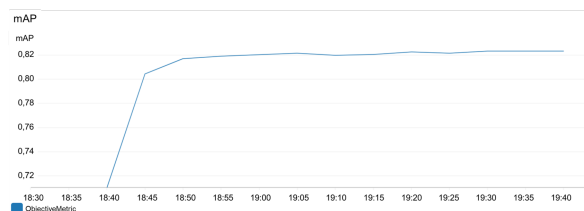


Fig. 7: Representació gràfica de l'entrenament respecte mAP en el temps

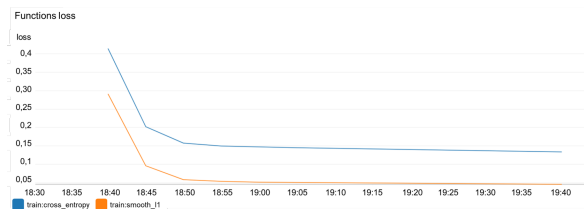


Fig. 8: Representació gràfica de l'entrenament respecte a les funcions loss en el temps

## 7.2 Resultats del model de detecció de mascaretes

El procés de creació del model de detecció de mascaretes és diferent a l'anterior. Per desenvolupar aquest model s'utilitza la configuració per defecte que aporta l'API i no requereix programació ni hiperparàmetres específics, com requereix el model de detecció de cares.

Per crear el model de detecció de mascaretes es necessita aportar un conjunt d'imatges (amb i sense mascareta).

Cal remarcar que aquest model es tracta d'un classificador, i per la creació del dataset, a diferència de l'anterior, no ha estat necessari fer *bounding boxes* ni tampoc l'etiquetatge manual de les imatges. El procés d'entrenament d'aquest model està completament automatitzat mitjançant l'API Rekognition.

Per obtenir el model de detecció de mascareta s'ha dividit el dataset en 1.100 imatges per l'entrenament i 276 de test.

Les mètriques resultants obtingudes han estat:

- F1 score: 1
- Average precision: 1
- Overall recall: 1

Els resultats obtinguts són molt bons. Aquest fet és degut, per una banda, a què el model ja està pre-entrenat, i amb l'addició de les dades aportades, el model aconsegueix més seguretat d'encertar. Per una altra banda, les imatges del dataset són cares en primer pla (com si fossin de carnet), fet que ajuda a que la classificació sigui molt més precisa.

## 7.3 Resultats de proves al Cloud

S'ha desenvolupat una sèrie de proves per cada API que es computa en el Cloud. Aquestes s'han realitzat en *scripts* en python locals. L'objectiu és comprovar de manera independent les funcions que s'utilitzen per a cada API.

TAULA 3: PROVES AL CLOUD

API	Nom de la prova	Definició
S3	Afegir objecte	Afegir un objecte a un bucket seleccionat
	Esborrar Objecte	Esborrar un objecte d'un bucket seleccionat
Rekognition	Crear col·lecció	Crear col·lecció d'imatges
	Afegir imatge a la col·lecció	S'afegeix la imatge d'un bucket a la col·lecció. Es guarda un identificador amb informació de la imatge
	Esborrar imatge de la col·lecció	S'esborra un element de la col·lecció seleccionant el seu ID corresponent
	Buscar imatge a la col·lecció	La dada d'entrada és una imatge del bucket. Es compara amb les imatges de la col·lecció i s'imprimeix, en cas que trobi una similitud, el seu ID. Aquest és la funció que fa possible el reconeixement facial
	Analitzar imatge amb etiquetes personalitzades	A partir del model de detecció de mascaretes entrenat prèviament, s'executa i permet saber si una imatge en local o una reproducció de vídeo en temps real, alguna persona porta mascareta
DynamoDB	Afegir ítem	Afegir un ítem a una taula seleccionada
	Actualitzar ítem	Actualitzar els atributs d'un ítem, identificant-lo a partir de la seva clau primària
Polly	Convertir text a audio	Escriure un text, i canviant la veu i idioma de l'àudio, guardar-lo de manera local i ser capaç de reproduir

En la taula 3 es mostra les proves realitzades per a cada API i quin tipus de prova s'ha realitzat. Cal destacar que totes les proves s'han desenvolupat satisfactòriament. Un cop se sap i es controla quines són les funcions necessàries i com s'han de cridar, s'ha procedit a realitzar el següent pas: la integració de tots els elements en un sol sistema.

## 7.4 Resultat del flux de dades

Un cop s'ha determinat que els dos models són funcionals i que les proves unitàries s'han realitzat amb èxit, es procedeix a integrar tots els elements. En aquesta subsecció s'exposa el flux de les dades del sistema un cop està completament integrat.

En el diagrama 9 s'observa el recorregut de les dades i les crides als serveis al llarg de tot el sistema.

Tot aquest procés es posa en marxa en el moment que el model de detecció de cares de la càmera en detecta una (*Face detection*). Es fa una fotografia i mitjançant una crida al model de detecció de mascaretes, es determina si la persona en porta o no. En el cas que no sigui així, el procediment torna a començar fins que l'usuari se la posi.

Un cop es fa la fotografia de la cara amb mascareta, aquesta es guarda a una carpeta d'objectes S3 (*DeepLens/*). Al realitzar-se aquesta acció, s'executa una funció Lambda (*rekognition-function*) que s'encarrega de fer el reconeixement de la persona, gràcies a les crides de la base de dades (*Users-registration*) i la col·lecció d'imatges (*faces-collection*) dels usuaris registrats. Un cop s'obté el nom de la persona, es guarda l'accés a una altra base de dades (*history*), i es converteix el seu nom en àudio i s'"envia" cap a la DeepLens. Allà es reproduceix i dona *feedback* per àudio que el reconeixement ha estat un èxit.

Prèviament a aquest procediment, s'han de registrar els usuaris. En afegir una fotografia d'una cara de l'usuari a la carpeta d'objectes S3 (*Users-registered/*), s'executa una funció Lambda (*upload-image-function*) que s'encarrega d'afegir la imatge a la col·lecció de cares (*faces-collection*), i el nom i identificador dins de la col·lecció a la base de dades (*Users-registration*).



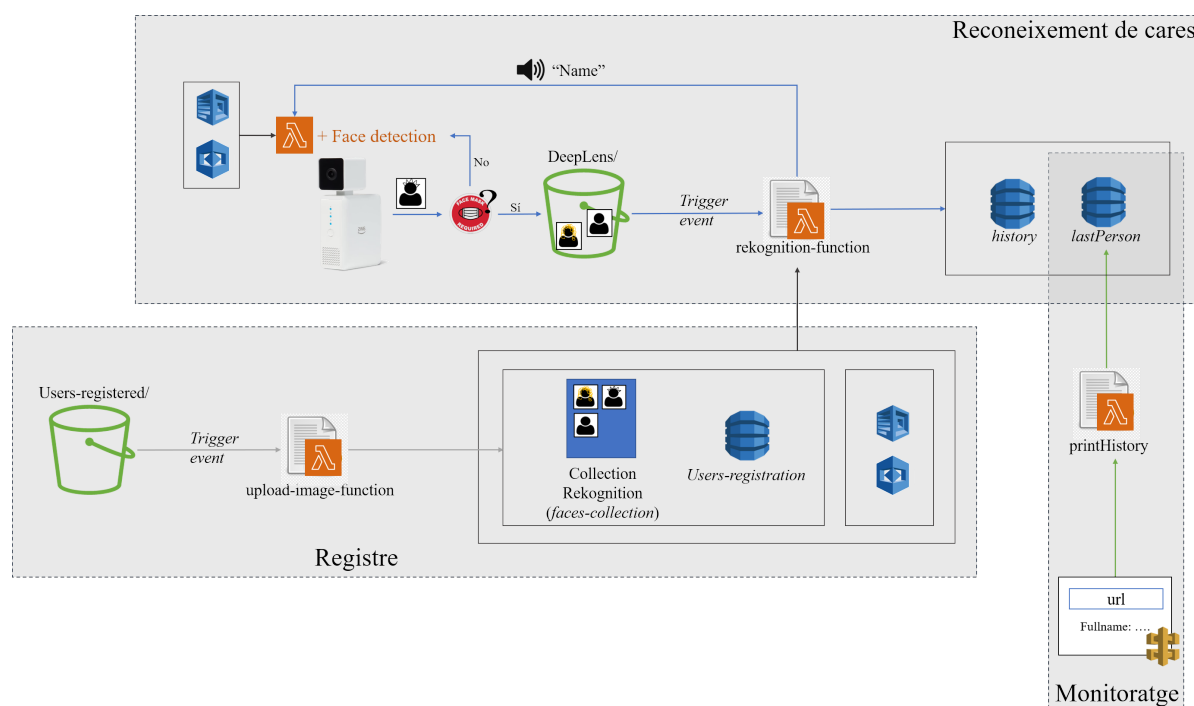


Fig. 9: Flux de dades

S'afegeix una funcionalitat extra que permet, mitjançant amb una *url*, accedir a l'última entrada de l'historial. Per realitzar aquesta funcionalitat, la funció que fa el reconeixement (*rekognition-function*) ha de guardar l'accés detectat a una altra base de dades (*lastPerson*), que només conté una entrada i aquesta es va actualitzant. Es crea una API amb el protocol REST, que a partir d'una funció Lambda (*printHistory*) escaneja el contingut d'aquesta base de dades i l'imprimeix per pantalla.

## 7.5 Estimació de costos

El Cloud és una infraestructura de lloguer de recursos, en la modalitat de pagament per ús, només es paga el que es consumeix. Està tabulat per uns conceptes que tenen una mètrica de consum específica. El preu unitari està definit per aquesta mètrica i usualment s'estableix per rangs d'ús.

AWS té una capa gratuïta que permet utilitzar aquests recursos fins a una capacitat limitada. Un cop se supera aquest lílindar, es carrega el cost de sobreutilització.

Tots els costos estan expressats en dòlars EEUU (\$).

### 7.5.1 Cost del prototip

Pel desenvolupament i proves d'aquest prototip s'han utilitzat recursos d'AWS tant gratuïts com de pagament. A la taula 4 es detalla la utilització i cost d'aquests serveis. Tota aquesta informació és obtinguda de la secció "billing" del Cloud.

### 7.5.2 Cost de la solució

Dels costos del prototip es determina dos tipus de cost: per cada accés i per la plataforma.

TAULA 4: COST DETALLAT DEL PROTOTIP

API	Funcions	Unitats processades	Cost/unitat	Cost/funció	Cost/API
S3	Requests-Tier1	2.000 requests	\$0,00	\$0,00	\$6,70
	PUT, COPY, POST, LIST	8.089 requests	\$0,005	\$0,04	
	Requests-Tier2	20.000 requests	\$0,00	\$0,00	
	GET i altres requests	16.654.391 requests	\$0,004	\$6,66	
Rekognition	USE1-FaceVectorsStored	1.7 Faces/month	\$0,00	\$0,00	\$0,812
	USE1-FaceVectorsStored	1.7 Faces/month	\$0,00	\$0,00	
	USE1-ImagesProcessed	87 imatges	\$0,00	\$0,00	
	Training CustomLabels	0,812 Hrs	\$1	\$0,812	
Lambda	Lambda-GB-Second	18.296 segons	\$0,00	\$0,00	\$0,00
	Request	37 requests	\$0,00	\$0,00	
DynamoDB	ReadCapacityUnit	1.648 ReadCapacityUnit-Hrs	\$0,00	\$0,00	\$0,00
	WriteCapacityUnit	1.648 WriteCapacityUnit-Hrs	\$0,00	\$0,00	
	TimedStorage-ByteHrs	0.000006 GB-Mo	\$0,00	\$0,00	
Polly	USE1-SynthesizeSpeech-Characters	1.975 Characters	\$0,00	\$0,00	\$0,00
Sagemaker	CreateLabelingJob	1000 objects	\$0,00	\$0,00	\$355,65
	CreateVolume-Gp2	2813 objects	\$0,08	\$225,04	
	Endpoint ML storage	0.006 GB-Mo	\$0,14	\$0,00	
	CreateVolume-Gp2	5.664 GB-Mo	\$0,14	\$0,79	
	Training Job ML storage	5.664 GB-Mo	\$0,14	\$0,79	
	RunInstance	0.523 Hrs	\$0,00	\$0,00	
	Host: ml.m4.xlarge	0.523 Hrs	\$0,00	\$0,00	
	RunInstance	83,177 Hrs	\$0,00	\$0,00	
	Notebook: ml.t3.medium	83,177 Hrs	\$0,00	\$0,00	
	RunInstance	33,939 Hrs	\$3,825	\$129,82	

En un accés hi intervenen serveis de procés, com el reconeixement i inserció a l'historial. Es fa la suposició que totes les persones estan prèviament registrades i porten la mascareta posada.

En la taula 5 es mostren els costos que se'n deriven quan es produeix un accés a l'edifici.

TAULA 5: COST D'UN ACCÉS

API	Funcions	Unitats processades	Cost/unitat	Cost/funció
S3	Requests-Tier1	1 request	\$0,005	\$0,005
	PUT, COPY, POST, LIST	1 request	\$0,005	\$0,005
	Requests-Tier2	1 request	\$0,004	\$0,004
	GET i altres requests	1 request	\$0,004	\$0,004
Rekognition	USE1-ImagesProcessed	2 imatges	\$0,001	\$0,002
	USE1-ImagesProcessed	2 imatges	\$0,001	\$0,002
Lambda	Request	2 requests	\$0,0000002	\$0,0000004
DynamoDB	WriteCapacityUnit	2 WriteCapacityUnit-Hrs	\$0,00000125	\$0,0000025
DynamoDB	WriteCapacityUnit	2 WriteCapacityUnit-Hrs	\$0,00000125	\$0,0000025
DynamoDB	GB-Month of storage (taula d'accessos)	100B	\$0,00	\$0,00
Polly	USE1-SynthesizeSpeech-Characters	50 Characters	\$0,000004	\$0,0002

El cost total d'un accés és: \$0,005 + \$0,004 + \$0,000023

$$+ \$0,002 + \$0,0000004 + \$0,0000025 + \$0,0000025 + \$0,00 + \$0,0002 = \$0,0112284 \text{ (cost}_{\text{accés}}).$$

En tots els càlculs es fa la suposició que un mes són 30 dies i cada persona accedeix 4 vegades al dia.

S'ha de tenir en compte el cas especial de la DynamoDB. Els primers 25GB emmagatzemats cada mes són gratuïts. A partir d'aquí, el cost és de \$0,25 per GB. Per superar aquest llindar s'haurien de fer 250 milions d'accessos (aproximadament 2,083 milions de persones). Per tant, es descarta l'opció de superar aquest llindar i no es té en compte aquest cost extra.

La següent fórmula indica el cost mensual d'accessos depenent el nombre de persones.

$$\begin{aligned} C_{\text{accessos}}(\text{persones}) &= \text{cost}_{\text{accés}} \cdot \text{num\_accessos/dia\_persona} \cdot \text{dies} \cdot \text{persones} \\ C_{\text{accessos}}(\text{persones}) &= 0,0112284 \cdot 4 \cdot 30 \cdot \text{persones} \\ C_{\text{accessos}}(\text{persones}) &= 1,35 \cdot \text{persones} \end{aligned}$$

A més, s'ha d'afegir a aquesta predicció, els costos de la plataforma (vegeu taula 6), és a dir, el preu que costa tenir la informació emmagatzemada i els models executant-se.

TAULA 6: COSTOS DE LA PLATAFORMA

API	Funcions	Unitats processades	Cost/unitat	Id
S3	GB-Month of storage	Fins 50TB	\$0,023	C1
Rekognition	Faces stored per month	Fins 1000 imatges	\$0,00	C2
		A partir de 1000 imatges	\$0,00001	C3
	CustomLabels Inference (Hrs)	1 Hr	\$4	C4
DynamoDB	GB-Month of storage	Fins 25 GB	\$0,00	C5

Continuant amb la mateixa premissa anterior, es calculen les següents funcions.

Funció per calcular el cost d'emmagatzemament del bucket dels registres d'usuaris de S3. Per superar el llindar gratuït, s'haurien de registrar 50 milions de persones, per la qual cosa es descarta el cost extra que suposaria superar el llindar. Aquest càlcul s'ha fet amb la consideració que cada persona ocupa 1MB (0,001 GB) en el bucket.

$$\begin{aligned} C_{S3}(\text{persones}) &= \text{mida\_imatge} \cdot C1 \cdot \text{persones} \\ C_{S3}(\text{persones}) &= 0,001 \cdot 0,023 \cdot \text{persones} \\ C_{S3}(\text{persones}) &= 0,000023 \cdot \text{persones} \end{aligned}$$

Funció per calcular el cost d'emmagatzemament de la taula dels registres d'usuaris de DynamoDB. Per superar el llindar gratuït, s'haurien de registrar 250 milions d'usuaris, per la qual cosa es descarta l'opció de superar el llindar. En aquest càlcul s'ha considerat que cada persona ocupa 100B (0,0000001 GB) de la taula.

$$\begin{aligned} C_{\text{dynamoDB}}(\text{persones}) &= \text{mida\_item} \cdot C5 \cdot \text{persones} \\ C_{\text{dynamoDB}}(\text{persones}) &= 0,0000001 \cdot 0,00 \cdot \text{persones} \\ C_{\text{dynamoDB}}(\text{persones}) &= 0 \end{aligned}$$

Funció per calcular el cost d'emmagatzemament de la col·lecció d'imatges de Rekognition:

si  $\text{persones} \leq 1.000$  :

$$\begin{aligned} C_{\text{Rek.Storage}}(\text{persones}) &= C2 \cdot \text{persones} \\ C_{\text{Rek.Storage}}(\text{persones}) &= 0,00 \cdot \text{persones} = 0 \end{aligned}$$

si  $\text{persones} > 1.000$  :

$$C_{\text{Rek.Storage}}(\text{persones}) = C3 \cdot (\text{persones} - 1.000)$$

$$C_{\text{Rek.Storage}}(\text{persones}) = 0,00001 \cdot (\text{persones} - 1.000)$$

Funció per calcular el cost del model de detecció de mascaretes. El cost, al contrari que la resta, es calcula respecte al temps i no persones.

$$\begin{aligned} C_{\text{Rek.Inference}}(\text{hores}) &= C4 \cdot \text{hores} \\ C_{\text{Rek.Inference}}(\text{hores}) &= 4 \cdot \text{hores} \end{aligned}$$

Per analitzar el cost mensual respecte el nombre d'empleats que s'autentiquen, es compara a la taula següent. Cal recordar que el nombre d'accessos s'ha considerat que és 4.

TAULA 7: COSTOS MENSUALS TOTALS SEGONS EL NOMBRE DE PERSONES

Persones	S3	DynamoDB	Rekognition Storage	Rekognition Inference	Accés	TOTAL
100	\$0,0023	\$0	\$0	\$2.880	\$135	\$3.015
1.000	\$0,023	\$0	\$0	\$2.880	\$1.350	\$4.230,02
10.000	\$0,23	\$0	\$0,09	\$2.880	\$13.500	\$16.380,32

S'observa que en nombre baix de persones, l'element més costós és el "Rekognition Inference" executat per la funció "CustomLabels Inference". Aquest es tracta del model de detecció de mascaretes que sempre està executant-se, encara que no es faci servir. La recomanació és que el model només estigui funcionant durant les hores que el sistema serà utilitzat. Per exemple, només funcionant de 8h a 20h es redueix el consum a la meitat.

En canvi, per a milers de persones, el cost depèn principalment del preu dels accessos, i no es veu tan influenciat pel cost del model de detecció de mascaretes.

Així doncs, a mesura que s'incrementa el nombre d'usuaris, el cost unitari per persona, baixa. D'altra banda, aquesta detecció deixarà de ser necessària un cop l'obligació de portar mascareta s'anul·li, el que resultarà un estalvi considerable en el cost, sobretot en nombre baix de persones.

## 8 CONCLUSIONS

En un cop d'ull es pot veure com el sistema és funcional. S'ha aconseguit dissenyar una arquitectura on l'usuari no ha de tocar cap element i només ha de seguir les indicacions que l'àudio de la càmera va dient. A més a més, comparant amb el sistema actual del Centre de Visió per Computador, s'ha realitzat una digitalització completa i permet escalabilitat en un futur, és a dir, es poden afegir noves funcionalitats sense realitzar grans canvis al disseny. Finalment, la seguretat d'accés millora considerablement, ja que serà molt difícil fer una suplantació d'usuaris.

Tanmateix, el sistema té una vulnerabilitat en aquest aspecte: dos bessons idèntics són considerats com el mateix individu, per tant, no els pot diferenciar. Una possible millora, per incrementar la seguretat, és afegir reconeixement de veu o analitzar altres autenticacions biomètriques per fer un sistema més precís i segur.

Encara que el model de detecció de cares dona resultats prou bons, té marge de millora. Per augmentar la precisió, s'hauria d'incrementar el dataset i invertir més hores a l'entrenament.

Respecte al punt de vista de l'usuari, el responsable de gestionar l'accés de les persones, s'ha fet una petita funcionalitat per veure des de qualsevol dispositiu l'última persona que ha entrat. L'altra interacció de l'usuari amb el sistema, és pel registre de nous membres, que actualment es fa de forma manual. Una posterior millora és dissenyar una aplicació web que permeti implementar aquestes funcionalitats, afegint-hi la seguretat requerida.

Després d'utilitzar la DeepLens durant el transcurs del projecte, s'ha arribat a la conclusió que, si bé la integració amb el Cloud està totalment definida i estandarditzada, aquesta aporta moltes limitacions, a diferència de desenvolupaments *open source*. Les funcionalitats utilitzades són les especificades per Amazon i no poden ser canviades. A més, en un principi es volia desplegar els dos models (detecció de cares i de mascaretes) a la càmera i que, d'aquesta manera s'executessin de forma local. Tanmateix, només permet un desplegament.

Finalment, encara que hi ha hagut limitació de hardware, l'eina és molt bona per ser utilitzada en situacions que necessitis una càmera funcionant contínuament i es vulgui treballar amb el Cloud d'Amazon. A més, AWS ha demostrat ser una eina de computació molt potent i satisfactòria, ja que posseeix molts models preentrenats i moltes aplicacions dins del ventall de visió per computador.

## 9 AGRAÏMENTS

Per acabar, agrair el suport de Dani Franco, Coen Antens i, sobretot d'en Toni Espinosa, que encara que estigués molt enfeinat, trobava el temps per respondre els meus dubtes i ajudant-me a avançar quan em trobava en un punt de bloqueig. També agrair a la meva família per la paciència d'haver estat escoltant sopar darrere sopar els avenços que anava fent, i especial menció al meu germà, ja que el vaig utilitzar de conillet d'índies diverses vegades.

## REFERÈNCIES

- [1] SafeHaven: seguridad en tiempo real (2018). Nathan Stone i Peter McLean. [consultat: 11 març 2021]. Disponible a Internet: <https://aws.amazon.com/es/deeplens/community-projects/SafeHaven/>
- [2] La primera cámara de video del mundo para desarrolladores compatible con aprendizaje profundo (2019). Amazon. [consultat: 11 març 2021]. Disponible a Internet: <https://aws.amazon.com/es/deeplens/>
- [3] AWS DeepLens Hardware (2021). Amazon. [consultat: 18 juny 2021]. Disponible a Internet: <https://docs.aws.amazon.com/deeplens/latest/dg/deeplens-hardware.html>
- [4] Reconocimiento facial: la tecnología para identificación de personas (2020). Grupo Atico34. [consultat: 14 juny 2021]. Disponible a Internet: <https://protecciondatos-lopdp.com/empresas/reconocimiento-facial/>
- [5] Full Face Recognition Statistics, Interesting Facts & New Trends (2020). Gigasource. [consultat: 14 juny 2021]. Disponible a Internet: <https://gigasource.io/face-recognition-statistics/>
- [6] Comparing Machine Learning as a Service: Amazon, Microsoft Azure, Google Cloud AI, IBM Watson (2021). Altexsoft. [consultat: 14 juny 2021]. Disponible a Internet: <https://www.altexsoft.com/blog/datascience/comparing-machine-learning-as-a-service-amazon-microsoft-azure-google-cloud-ai-ibm-watson/>
- [7] Amazon Rekognition (2021). Amazon Web Services. [consultat: 4 abril 2021]. Disponible a Internet: <https://aws.amazon.com/es/rekognition/?blog-cards.sort-by=item.additionalFields.createdAt&blog-cards.sort-order=desc>
- [8] AWS DeepLens Sample Projects Overview (2021). Amazon. [consultat: 29 maig 2021]. Disponible a Internet: <https://docs.aws.amazon.com/deeplens/latest/dg/deeplens-templated-projects-overview.html>
- [9] Object Detection Algorithm (2021). Amazon. [consultat: 14 juny 2021]. Disponible a Internet: <https://docs.aws.amazon.com/sagemaker/latest/dg/object-detection.html>
- [10] AWS IoT Greengrass (2020). Amazon Web Services. [consultat: 22 març 2021]. Disponible a Internet: <https://aws.amazon.com/es/greengrass/>
- [11] Crear e implementar un proyecto de AWS DeepLens (2019). Amazon. [consultat: 20 març 2021]. Disponible a Internet: <https://aws.amazon.com/es/getting-started/hands-on/create-deploy-project-deeplens/>
- [12] OpenVINO™ Toolkit Overview (2021). OpenVINO. [consultat: 14 juny 2021]. Disponible a Internet: <https://docs.openvinotoolkit.org/2021.1/index.html>
- [13] Converting a MXNet\* Model (2021). OpenVINO. [consultat: 14 juny 2021]. Disponible a Internet: [https://docs.openvinotoolkit.org/latest/openvino\\_4\\_0\\_CS\\_M0\\_DG\\_prepare\\_model\\_convert\\_model\\_convert\\_model\\_from\\_mxnet.html](https://docs.openvinotoolkit.org/latest/openvino_4_0_CS_M0_DG_prepare_model_convert_model_convert_model_from_mxnet.html)
- [14] ¿Qué es y para qué sirve el Cloud Computing? (2016). Juan Hernandez. [consultat: 30 maig 2021]. Disponible a Internet: <http://blog.editaFacil.es/que-es-y-para-que-sirve-el-cloud-computing/#:~:text=La%20computaci%C3%B3n%20en%20la%20nube,tu%20casa%20o%20tu%20oficina.>
- [15] mAP (mean Average Precision) might confuse you! (2021). Towards Data Science. [consultat: 14 juny 2021]. Disponible a Internet: <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bf9a9e2>